

KALAMIA

Jurnal Sains dan Teknologi

Senyawa Antikanker dari Aktinomisetes Laut
(Esti Wahyu Widowati)

Aplikasi Algoritma Genetika untuk Menyelesaikan Masalah
Travelling Salesman Problem (TSP)
(Muhammad Abrori, Endang Sulistyowati, dan Sakinatul Chasanah)

Sistem Pemesanan Buku Online Interaktif dengan Menggunakan
Library JQuery
(Ahmad 'Athaullah, Agus Mulyanto)

Energi Ikat Deuteron: Suatu Tinjauan Teoritis dan Eksperimen
(Sri Purwaningsih)

Usabilty Sebagai Pengukur Tingkat Kepuasan
Pengguna Sistem Aplikasi Komputer
(Ade Ratnasari)

DAFTAR ISI

Senyawa Antikanker dari Aktinomisetes Laut

Esti Wahyu Widowati 1

Aplikasi Algoritma Genetika untuk Menyelesaikan Masalah *Travelling Salesman Problem* (Tsp)

Muchammad Abrori, Endang Sulistyowati, dan Sakinatul Chasanah 15

Sistem Pemesanan Buku Online Interaktif dengan Menggunakan *Library JQuery*

Ahmad ` Athaullah, Agus Mulyanto 32

Energi Ikat Deuteron: Suatu Tinjauan Teoritis dan Eksperimen

Sri Purwaningsih 43

***Usability* Sebagai Pengukur Tingkat Kepuasan Pengguna Sistem Aplikasi Komputer**

Ade Ratnasari 50

KAUNIA

Jurnal Sains dan Teknologi

Vol. VII No. 1 April 2011 / 1432 H

Penanggung Jawab

Prof. Drs. H. Akh. Minhaji, M.A., Ph.D.

Redaktur

Kifayah Amar, Ph.D.

Sekretaris Redaksi

Anti Damayanti, S.Si., M.Bio.Mol

Liana Aisyah, S.Si., M.A.

Penyunting/Editor

Nurochman, S.Kom., M.Kom.

Nita Handayani, S.Si., M.Si.

Staf Sekretaris Redaksi

Latifah

Robi'atul Chalimah, SIP

Penyunting Ahli

Dr. Moedji Raharto

(Institut Teknologi Bandung)

Dr. Farchani Rosyid

(Universitas Gadjah Mada)

Prof. Drs. Subanar, Ph.D.

(Universitas Gadjah Mada)

Dr. Khabib Musthofa

(Universitas Gadjah Mada)

Dr. Tjut Djohan Sugandawati, M.Sc.

Dr. Indah Emilia

(Universitas Gadjah Mada)

Dr. Budi Daryono

(Universitas Gadjah Mada)

Dr. Subagyo

(Universitas Gadjah Mada)

Dr. Kuwat Triyana

(Universitas Gadjah Mada)

Dr. Marsigit, M.Pd.

(Universitas Negeri Yogyakarta)

Prof. Dr. Susiknan Azhari

(UIN Sunan Kalijaga)

Terbit Pertama Kali

April 2005

Frekuensi Terbit

2 (dua) kali setahun

Alamat Redaksi

Fakultas Sains dan Teknologi

UIN Sunan Kalijaga Yogyakarta

Jalan Marsda Adi Sucipto Yogyakarta 55281 Indonesia

Telp. : +62-274-519739; Fax : +62-274-540971

Email: kifayah@uin-suka.ac.id

APLIKASI ALGORITMA GENETIKA UNTUK MENYELESAIKAN MASALAH *TRAVELLING SALESMAN PROBLEM* (TSP)

Muchammad Abrori, Endang Sulistyowati, dan Sakinatul Chasanah

Program Studi Matematika Fakultas Sains dan Teknologi,
UIN Sunan Kalijaga

Abstract

Travelling Salesman Problems (TSP) is a classical optimization problem which refers to the directed graph that search Hamilton circuit with the shortest distance or minimum travel tour. TSP is categorized as a NP-Hard problem that can be solved perfectly with polynomial time. The TSP problem is generally solved using heuristic method. Heuristic method, simpler than other methods, can solve complex problems with more varied results and shorter calculation time. We use Genetic Algorithm, a kind of heuristic method, to determine the shortest route through every city only once and return to original departure city. Genetic Algorithm is an analogy form of Darwin's Evolution Theory and Principles of Genetics in Biological Science. This algorithm is very effective to solve the complex optimization problems that is difficult to be solved with conventional method. The solution obtained by genetic algorithm depends on type of operator and parameter used in the algorithm. Genetic operator includes selection, crossover and mutation. The parameters involve population size, number of generations, crossover probability and mutation probability. Genetic Algorithm as heuristics algorithm can search the solution for TSP problem, but because it starts from a random solution then the result is not always the optimal solution.

Keywords: *Travelling Salesman Problem (TSP), Genetic Algorithm, Heuristic.*

A. Pendahuluan

Dalam kehidupan, sering dilakukan perjalanan dari satu tempat atau kota ke tempat yang lain dengan mempertimbangkan efisiensi waktu dan biaya sehingga diperlukan ketepatan dalam menentukan jalur terpendek yang melintasi kota-kota tersebut. Hasil penentuan jarak terpendek akan menjadi pertimbangan dalam pengambilan keputusan untuk menunjukan rute perjalanan yang akan ditempuh (Bambang, 2008).

Salah satu masalah mengenai rute terpendek adalah masalah *Travelling Salesman Problem*. *Travelling Salesman Problem (TSP)* merupakan suatu masalah yang cukup terkenal pada teori graf. Secara umum deskripsi dari masalah TSP adalah:

‘Diberikan sejumlah kota dan jarak antar kota. Tentukan sirkuit terpendek yang harus dilakukan oleh seorang pedagang bila pedagang itu berangkat dari sebuah kota asal dan menyinggahi setiap kota tepat satu kali dan kembali

lagi ke kota asal keberangkatan. Kota dapat dinyatakan sebagai simpul graf, sedangkan sisi menyatakan jalan yang menghubungkan antar dua buah kota. Bobot pada sisi menyatakan jarak antara dua buah kota.’(Rinaldi Munir, 2001)

Masalah TSP adalah masalah menentukan sirkuit Hamilton yang memiliki bobot minimum pada sebuah graf terhubung. TSP ini digolongkan ke dalam *NP-Hard*. Dari pendekatan di atas, maka metode yang digunakan adalah Heuristik. Heuristik merupakan suatu metode pendekatan untuk memecahkan suatu masalah dan tidak peduli apakah solusinya bisa dibuktikan benar atau tidak. Akan tetapi secara umum, heuristic dapat menyelesaikan masalah dengan lebih sederhana. Masalah yang lebih kompleks dapat diselesaikan dengan hasil yang lebih variatif dan waktu perhitungannya lebih singkat (Widhiyasa, 2007).

Metode Heuristik terdiri dari beberapa macam Algoritma yang biasa digunakan, seperti *Hill Climbing*, *Ant Colony*, *Simulated Annealing*, *Branch and Bound*, *Tabu Search*, *Nearest Neighbor* dan Algoritma Genetika.

Algoritma Genetika adalah suatu bentuk analogi dari Teori Evolusi Darwin dan Prinsip Genetika dalam Ilmu Biologi. Teori ini menerangkan bahwa evolusi disebabkan oleh adanya variasi genetik dan proses seleksi alam pada setiap generasi yang muncul. Seleksi alam dapat diartikan sebagai seleksi terhadap individu-individu yang hidup di alam atau lingkungannya.

Penelitian ini bertujuan untuk mengetahui penyelesaian masalah TSP dengan menggunakan Algoritma Genetika. Dengan mengetahui gambaran aplikasi Algoritma Genetika dalam masalah optimasi khususnya masalah TSP, Algoritma Genetika dapat dimanfaatkan untuk membantu perusahaan yang bergerak di bidang transportasi, seperti perusahaan bus, perusahaan travel dan jasa pengiriman, dalam menentukan rute terpendek. Dengan demikian, perusahaan tersebut dapat menghemat waktu, tenaga dan biaya.

B. Metode Penelitian

Tahapan yang akan dilakukan dalam penelitian ini adalah:

1. Mempelajari tentang Algoritma Genetika dan TSP.
2. Mempelajari penggunaan Algoritma Genetika dalam menentukan rute terpendek dari beberapa kota pada masalah TSP dengan menentukan jaraknya terlebih dahulu.
3. Mengerjakan contoh masalah TSP dengan Algoritma Genetika dimulai dari inisialisasi populasi awal, evaluasi, melakukan proses seleksi, *crossover* dan mutasi.

C. Travelling Salesman Problem (TSP)

TSP merupakan salah satu permasalahan penting dalam dunia matematika. Persoalan TSP merupakan contoh lain dari persoalan optimasi klasik yang mengacu pada graf berarah. Pada persoalan TSP ini terdapat sejumlah kota dan jarak antara kota-kota tersebut. Yang menjadi masalah adalah sirkuit mana yang harus dicari oleh pedagang tersebut bila dia berangkat dari sebuah kota asal dan menyinggahi setiap kota tepat satu kali dan kembali lagi ke kota asal keberangkatan. Solusi dari TSP ialah lintasan yang dilalui oleh salesman tersebut. Tentunya solusi terbaik atau optimal dari permasalahan ini ialah lintasan dengan jarak terpendek atau disebut juga dengan *tour* perjalanan minimum. Model TSP dinyatakan dalam bentuk graf. Dengan kata lain, TSP termasuk ke dalam masalah menentukan lintasan atau sirkuit Hamilton.

Dalam graf, kota dinyatakan dengan titik, sedangkan sisi menyatakan jalan yang menghubungkan antara dua kota. Persoalan TSP ini adalah menentukan sirkuit Hamilton yang memiliki bobot terkecil pada sebuah graf terhubung (Herlianti, 2005).

TSP tidak hanya berlaku untuk graf lengkap, akan tetapi berlaku juga untuk graf tidak lengkap asalkan memiliki sirkuit Hamilton. Persoalan TSP merupakan persoalan yang sulit jika dipandang dari sudut komputasi. Secara teoritis, TSP dapat dipecahkan dengan mengenumerasi $(n - 1)!/2$ buah sirkuit Hamilton. Misalnya untuk $n = 6$, dengan n adalah jumlah kota, maka akan ada 60 buah penyelesaian.

D. Aplikasi Algoritma Genetika pada TSP

1. Inisialisasi Populasi Awal

Representasi data yang digunakan pada Algoritma Genetika untuk masalah TSP adalah representasi bilangan bulat (*integer*). Data disajikan dalam bentuk rangkaian barisan bilangan bulat, dimana satu rangkaian mempresentasikan individu yang disebut dengan kromosom. Kromosom terdiri dari kumpulan gen yang berupa bilangan bulat. Gen dalam kromosom tersebut mempresentasikan kota yang dikunjungi sales dan posisi gen mempresentasikan posisi kunjungan. Dengan demikian, kromosom tersebut mempresentasikan rute perjalanan yang ditempuh sales. Artinya, solusi yang diberikan adalah urutan kota yang dilalui sales. Urutan kota yang dikunjungi harus berbeda satu sama lain. Jenis pengkodean yang paling tepat untuk solusi di atas adalah pengkodean permutasi (Fitria, dkk, 2005).

2. Evaluasi

Evaluasi dilakukan dengan menghitung jarak antar kota pada rute yang telah tersusun dalam kromosom. Karena tiap *locus* berisi nomor yang mewakili tiap kota, maka perhitungan nilai *fitness* dilakukan dengan menjumlahkan jarak antara kota asal dengan *locus* pertama, kota pada *locus* pertama dengan *locus* kedua, dan selanjutnya sampai kembali lagi pada kota asal.

Proses evaluasi melakukan pencarian jarak antara *gen* lokasi sekarang dan selanjutnya, kemudian menambahkannya ke jarak awal dari tiap kromosom yang telah dibangkitkan.

3. Seleksi kromosom

Seleksi kromosom dilakukan agar hanya kromosom-kromosom yang berkualitas yang dapat melanjutkan ke proses Algoritma Genetika berikutnya. Pada proses penyeleksian digunakan parameter *fitness* untuk menentukan seberapa baik kromosom akan bertahan hidup. Masing-masing rute pada populasi awal dihitung jarak, nilai *fitness*, probabilitas *fitness* dan probabilitas kumulatif *fitness*nya. Tahap-tahap perhitungan *fitness* adalah sebagai berikut:

1. Mencari jarak tempuh tiap rute (f_i)
2. Mencari nilai *fitness* tiap rute (fit_i)

$$fit_i = \frac{1}{f_i} \quad (3.1)$$

3. Mencari total *fitness* ($\sum_{i=1}^N fit_i$)

4. Mencari probabilitas *fitness* tiap rute (p_i)

$$p_i = \frac{fit_i}{\sum_{i=1}^N fit_i} \quad (3.3)$$

5. Mencari probabilitas kumulatif tiap rute (q_i)

$$q_i = \sum_k^i p_k \quad (3.4)$$

6. Bangkitkan bilangan random antar 0 sampai 1.
7. Pemilihan rute populasi berikutnya dilakukan dengan membandingkan bilangan random dengan probabilitas kumulatif tiap rute (q_i), dimana $r(k) < q_i$, maka kromosom ke- n yang terpilih (Obitko, 1998).

4. Crossover

Proses *crossover* dilakukan untuk menghasilkan keturunan dari dua *parent* yang terpilih pada proses *crossover*. Keturunan yang dihasilkan merupakan kombinasi gen-gen yang dimiliki oleh kromosom kedua *parent*-nya. Karena yang digunakan adalah

pengkodean permutasi, maka untuk *crossover* akan digunakan *crossover* satu titik dengan operator *Order Crossover* (OX), untuk menjaga konsistensi urutan nilai pada kromosom.

Sebelum dilakukan *crossover*, terlebih dahulu dipilih kromosom-kromosom yang akan melakukan *crossover*. Caranya adalah membangkitkan bilangan random r dari 0 sampai 1 sebanyak jumlah populasi, kemudian bandingkan dengan nilai probabilitas *crossover* (p_c). Jika $r_k < p_c$ maka kromosom tersebut akan dipilih menjadi induk. Kemungkinan rute yang terlibat dalam proses *crossover* sebanyak $p_c \cdot$ jumlah populasi.

Kromosom-kromosom melakukan *crossover* secara berpasangan. Jika kromosom yang terpilih untuk melakukan *crossover* berjumlah ganjil, maka satu kromosom yang terpilih tersebut dihapus.

Langkah – langkah yang dilakukan pada proses *crossover* adalah:

1. Tentukan probabilitas *crossover*
2. Bangkitkan bilangan random r dari 0 sampai 1 sebanyak jumlah populasi.

$$\text{Bil Random}_n = \text{random}(0,1) \tag{3.5}$$

3. Jika *parent* yang terpilih jumlahnya hanya satu maka proses diulang sampai jumlah *parent* lebih dari 1.
4. Jika $\text{Bil Random}_n < P_c$, maka kromosom ke- n terpilih untuk melakukan *crossover*.
5. Bangkitkan bilangan random dari 1 sampai $L-1$ (L =panjang kromosom) sampai sebanyak kromosom induk yang akan di-*crossover*.
6. Untuk *offspring*, salin gen *parent* 1 dari awal sampai titik persilangan dan lihat satu per satu gen pada parent 2. Jika gen belum ada, maka tambahkan gen tersebut pada *offspring*.

5. Mutasi

Karena pengkodean yang digunakan adalah pengkodean permutasi, maka skema mutasi yang digunakan adalah *swap mutation* yaitu dengan menukarkan 2 nilai gen yang berurutan pada sebuah kromosom.

Sebelum dilakukan mutasi, terlebih dahulu ditentukan panjang seluruh gen yang akan melakukan mutasi sebanyak jumlah gen dalam satu kromosom * jumlah kromosom. Pemilihan gen-gen yang akan dimutasi dilakukan dengan membangkitkan bilangan acak antara 1 sampai panjang total gen. Sedangkan jumlah gen yang akan dimutasi diperoleh dari P_m * panjang seluruh gen. Posisi gen yang akan dimutasi diperoleh dengan membangkitkan bilangan acak sebanyak jumlah gen yang akan dimutasi. Kemudian dilakukan proses mutasi.

Langkah – langkah yang dilakukan pada proses mutasi yaitu:

1. Hitung panjang total gen dalam satu populasi

$$\text{Panj tot gen} = \text{jum gen satu kromosom} * \text{jum kromosom} \quad (3.6)$$

2. Tentukan probabilitas mutasi.
3. Tentukan jumlah gen yang akan dimutasi

$$\text{Jum gen} = P_m * \text{Panj tot gen} \quad (3.7)$$

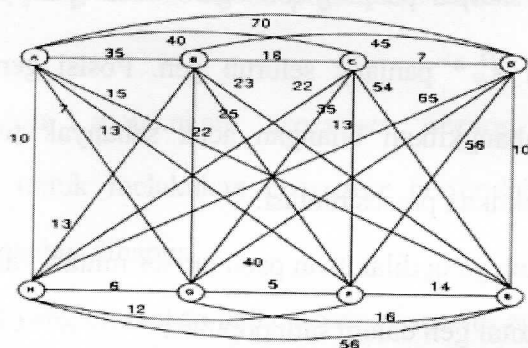
4. Bangkitkan bilangan random dari 1 sampai panjang total gen sebanyak jumlah gen yang akan dimutasi.
5. Tukarkan posisi gen sesuai bilang random yang diperoleh.

E. Pembahasan

Contoh kasus: Seorang sales akan melakukan perjalanan ke 8 kota, yaitu kota A, B, C, D, E, F, G dan H. Perjalanan akan dimulai dari kota A dan kembali lagi ke kota A. Jarak antar kota masing-masing terlihat seperti gambar di bawah ini:

	A	B	C	D	E	F	G	H
A	0	35	40	70	15	13	7	10
B	35	0	18	45	23	25	22	13
C	40	18	0	7	54	13	35	22
D	70	45	7	0	10	56	40	65
E	15	23	54	10	0	14	16	56
F	13	25	13	56	14	0	5	12
G	7	22	35	40	16	5	0	6
H	10	13	22	65	56	12	6	0

Jika ditampilkan dalam bentuk graf :



Gambar 1. Rute yang dapat dilalui untuk 8 Kota pada Kasus TSP

Masalahnya: bagaimana rute perjalanan yang harus dilalui sales agar setiap kota dapat dilewati dengan jarak minimum.

a. Inisialisasi populasi awal

Teknik pengkodean untuk masalah ini sama dengan contoh 1, yaitu pengkodean permutasi. Setelah menentukan teknik pengkodean dan sebelum ke langkah Algoritma Genetika selanjutnya, terlebih dahulu ditentukan kriteria berhenti, yaitu setelah dalam beberapa generasi berturut-turut diperoleh nilai *fitness* terendah yang tidak berubah. Pemilihan nilai *fitness* yang terendah digunakan sebagai syarat karena nilai tersebut mempresentasikan jarak terdekat yang dicari pada persoalan TSP ini. Dalam kasus ini terdapat 7 kota yang akan menjadi gen dalam kromosom, yaitu kota selain kota asal.

Dalam kasus ini akan digunakan 10 buah populasi yang diambil secara acak dalam satu generasi. Maka akan diciptakan 10 buah kromosom secara acak, yaitu:

Kromosom [1] = [GFEDCBH]

Kromosom [2] = [FGHBCDE]

Kromosom [3] = [HFEDCBG]

Kromosom [4] = [EDCBHGF]

Kromosom [5] = [BHFCDEG]

Kromosom [6] = [CDEFGHB]

Kromosom [7] = [DEBCHGF]

Kromosom [8] = [FBCDEGH]

Kromosom [9] = [GHBCDEF]

Kromosom [10] = [EGHFCDB]

b. Evaluasi Kromosom

Evaluasi kualitas kromosom dilakukan dengan menghitung nilai fungsi tujuan (f_i) dari tiap kromosom yang telah dibangkitkan. Nilai tujuan pada masalah di atas didapat dengan menghitung jarak antar kota yang sesuai dengan urutan dalam kromosom.

$$f_1 = AG + GF + FE + ED + DC + CB + BH + HA$$

$$f_1 = 7 + 5 + 14 + 10 + 7 + 18 + 13 + 10 = 84$$

$$f_2 = AF + FG + GH + HB + BC + CD + DE + EA$$

$$f_2 = 13 + 5 + 6 + 13 + 18 + 7 + 10 + 15 = 87$$

$$f_3 = AH + HF + FE + ED + DC + CB + BG + GA$$

$$f_3 = 10 + 12 + 14 + 10 + 7 + 18 + 22 + 7 = 100$$

$$f_4 = AE + ED + DC + CB + BH + HG + GF + FA$$

$$f_4 = 15 + 10 + 7 + 18 + 13 + 6 + 5 + 13 = 87$$

$$f_5 = AB + BH + HF + FC + CD + DE + EG + GA$$

$$f_5 = 35 + 13 + 12 + 13 + 7 + 10 + 16 + 7 = 113$$

$$f_6 = AC + CD + DE + EF + FG + GH + HB + BA$$

$$f_6 = 40 + 7 + 10 + 14 + 5 + 6 + 13 + 35 = 130$$

$$f_7 = AD + DE + EB + BC + CH + HG + GF + FA$$

$$f_7 = 70 + 10 + 23 + 18 + 22 + 6 + 5 + 13 = 167$$

$$f_8 = AF + FB + BC + CD + DE + EG + GH + HA$$

$$f_8 = 13 + 25 + 18 + 7 + 10 + 16 + 6 + 10 = 105$$

$$f_9 = AG + GH + HB + BC + CD + DE + EF + FA$$

$$f_9 = 7 + 6 + 13 + 18 + 7 + 10 + 14 + 13 = 88$$

$$f_{10} = AE + EG + GH + HF + FC + CD + DB + BA$$

$$f_{10} = 15 + 16 + 6 + 12 + 13 + 7 + 45 + 35 = 149$$

c. Seleksi kromosom

Karena masalah TSP bertujuan meminimalkan jarak, maka kromosom dengan nilai *fitness* yang lebih kecil akan mempunyai probabilitas yang lebih besar untuk terpilih kembali. Oleh karena itu, digunakan *invers* dari *fitness*-nya, dengan rumus:

$$fit_i = \frac{1}{f_i}$$

$$fit_1 = \frac{1}{f_1} = \frac{1}{84} = 0,0119$$

$$fit_6 = \frac{1}{f_6} = \frac{1}{130} = 0,0076$$

$$fit_2 = \frac{1}{f_2} = \frac{1}{87} = 0,0114$$

$$fit_7 = \frac{1}{f_7} = \frac{1}{167} = 0,0059$$

$$fit_3 = \frac{1}{f_3} = \frac{1}{100} = 0,01$$

$$fit_8 = \frac{1}{f_8} = \frac{1}{105} = 0,0095$$

$$fit_4 = \frac{1}{f_4} = \frac{1}{87} = 0,0114$$

$$fit_9 = \frac{1}{f_9} = \frac{1}{88} = 0,0113$$

$$fit_5 = \frac{1}{f_5} = \frac{1}{113} = 0,0088$$

$$fit_{10} = \frac{1}{f_{10}} = \frac{1}{149} = 0,0067$$

Total *fitness* semua individu adalah:

$$T(fit_i) = \sum_{i=1}^N fit_i$$

$$T(fit_i) = fit_1 + fit_2 + fit_3 + fit_4 + fit_5 + fit_6 + fit_7 + fit_8 + fit_9 + fit_{10}$$

$$T(\text{fit}_i) = 0,0119 + 0,0114 + 0,01 + 0,0114 + 0,0088 + 0,0076 + 0,0059 + 0,0095 + 0,0113 + 0,0067 = 0,0946$$

Setelah diperoleh *fitness* dan total *fitness*, probabilitas masing-masing individu dihitung dengan menggunakan rumus:

$$p_i = \frac{\text{fit}_i}{T(\text{fit}_i)}$$

$$p_1 = \frac{\text{fit}_1}{T(\text{fit}_i)} = \frac{0,0119}{0,0946} = 0,1257$$

$$p_6 = \frac{\text{fit}_6}{T(\text{fit}_i)} = \frac{0,0076}{0,0946} = 0,0803$$

$$p_2 = \frac{\text{fit}_2}{T(\text{fit}_i)} = \frac{0,0114}{0,0946} = 0,1205$$

$$p_7 = \frac{\text{fit}_7}{T(\text{fit}_i)} = \frac{0,0059}{0,0946} = 0,0623$$

$$p_3 = \frac{\text{fit}_3}{T(\text{fit}_i)} = \frac{0,01}{0,0946} = 0,1057$$

$$p_8 = \frac{\text{fit}_8}{T(\text{fit}_i)} = \frac{0,0095}{0,0946} = 0,1004$$

$$p_4 = \frac{\text{fit}_4}{T(\text{fit}_i)} = \frac{0,0114}{0,0946} = 0,1205$$

$$p_9 = \frac{\text{fit}_9}{T(\text{fit}_i)} = \frac{0,0113}{0,0946} = 0,1194$$

$$p_5 = \frac{\text{fit}_5}{T(\text{fit}_i)} = \frac{0,0088}{0,0946} = 0,0930$$

$$p_{10} = \frac{\text{fit}_{10}}{T(\text{fit}_i)} = \frac{0,0067}{0,0946} = 0,0708$$

Dari probabilitas di atas, terlihat bahwa kromosom ke-1 mempunyai *fitness* paling kecil sehingga kromosom ke-1 mempunyai probabilitas yang lebih besar dibandingkan kromosom lainnya untuk terpilih pada generasi selanjutnya.

Untuk proses seleksi menggunakan *roulette-wheel*, terlebih dahulu dicari nilai kumulatif dari probabilitasnya.

$$q_i = \sum_k^i p_k$$

$$q_1 = p_1 = 0,1257$$

$$q_2 = p_1 + p_2 = 0,1257 + 0,1205 = 0,2462$$

$$q_3 = p_1 + p_2 + p_3 = 0,1257 + 0,1205 + 0,1057 = 0,3519$$

$$q_4 = p_1 + p_2 + p_3 + p_4 = 0,1257 + 0,1205 + 0,1057 + 0,1205 = 0,4724$$

$$q_5 = p_1 + p_2 + p_3 + p_4 + p_5$$

$$q_5 = 0,1257 + 0,1205 + 0,1057 + 0,1205 + 0,0930 = 0,5654$$

$$q_6 = p_1 + p_2 + p_3 + p_4 + p_5 + p_6$$

$$q_6 = 0,1257 + 0,1205 + 0,1057 + 0,1205 + 0,0930 + 0,0803 = 0,6457$$

$$q_7 = p_1 + p_2 + p_3 + p_4 + p_5 + p_6 + p_7$$

$$q_7 = 0,1257 + 0,1205 + 0,1057 + 0,1205 + 0,0930 + 0,0803 + 0,0623 = 0,7080$$

$$q_8 = p_1 + p_2 + p_3 + p_4 + p_5 + p_6 + p_7 + p_8$$

$$q_8 = 0,1257 + 0,1205 + 0,1057 + 0,1205 + 0,0930 + 0,0803 + 0,0623 + 0,1004 = 0,8084$$

$$q_9 = p_1 + p_2 + p_3 + p_4 + p_5 + p_6 + p_7 + p_8 + p_9$$

$$q_9 = 0,1257 + 0,1205 + 0,1057 + 0,1205 + 0,0930 + 0,0803 + 0,0623 + 0,1004 + 0,1194 = 0,9278$$

$$q_{10} = p_1 + p_2 + p_3 + p_4 + p_5 + p_6 + p_7 + p_8 + p_9 + p_{10}$$

$$q_{10} = 0,1257 + 0,1205 + 0,1057 + 0,1205 + 0,0930 + 0,0803 + 0,0623 + 0,1004 + 0,1194 + 0,0708 = 0,9986$$

Proses *roulette-wheel* adalah membangkitkan nilai acak r antara 0-1.

$$r_1 = 0,9501 \quad r_6 = 0,7621$$

$$r_2 = 0,2311 \quad r_7 = 0,4565$$

$$r_3 = 0,6068 \quad r_8 = 0,0185$$

$$r_4 = 0,4860 \quad r_9 = 0,8214$$

$$r_5 = 0,8913 \quad r_{10} = 0,4447$$

Jika $r_k < q_k$, maka kromosom ke- k sebagai induk. Kemudian, *roulette-wheel* diputar sebanyak jumlah kromosom, yaitu 10 kali. Dari bilangan random di atas, akan dihasilkan populasi baru hasil seleksi, yaitu:

Kromosom [1] = [GFEDCBH]

Kromosom [2] = [DEBCHGF]

Kromosom [3] = [EDCBHGF]

Kromosom [4] = [FBCDEGH]

Kromosom [5] = [HFEDCBG]

Kromosom [6] = [CDEFGHB]

Kromosom [7] = [BHFCDEG]

Kromosom [8] = [GHBCDEF]

Kromosom [9] = [EGHFCDB]

Kromosom [10] = [GFEDCBH]

d. Perkawinan silang/*crossover*

Crossover pada TSP ini diimplementasikan dengan skema *order crossover*. Pada skema ini, satu bagian kromosom dipertukarkan dengan tetap menjaga urutan kota yang bukan bagian dari kromosom tersebut. Kromosom yang dijadikan induk dipilih secara acak dan jumlah kromosom yang di *crossover* dipengaruhi oleh parameter *crossover probability* (pc).

Misalkan, kita tentukan $pc = 60\%$. Berarti, jika bilangan random yang dihasilkan < 0.6 , maka kromosom tersebut akan dipilih menjadi induk baru.

Pertama, kita bangkitkan bilangan acak r sebanyak jumlah populasi, yaitu 10 kali.

$$r_1 = 0,6154 \quad r_6 = 0,4057$$

$$r_2 = 0,7919 \quad r_7 = 0,9355$$

$$r_3 = 0,9218 \quad r_8 = 0,9169$$

$$r_4 = 0,7382 \quad r_9 = 0,4103$$

$$r_5 = 0,1763 \quad r_{10} = 0,8936$$

Kromosom ke- k dipilih sebagai induk jika $r_k < pc$. Maka, yang akan dijadikan induk adalah kromosom[5], kromosom[6], kromosom[9].

Setelah melakukan pemilihan induk, proses selanjutnya adalah menentukan posisi *crossover*, yaitu dengan membangkitkan bilangan acak antara 1 sampai dengan panjang kromosom-1. Dalam kasus TSP ini, bilangan acaknya adalah antara 1-6. Misal diperoleh bilangan acaknya, maka gen yang ke-1 pada kromosom induk pertama diambil kemudian ditukar dengan gen pada kromosom induk kedua yang belum ada pada induk pertama dengan tetap memperhatikan urutannya. Bilangan acak untuk 3 kromosom induk yang akan di- *crossover*:

$$q_5 = 2$$

$$q_6 = 4$$

$$q_9 = 1$$

Jadi populasi setelah dilakukan *crossover*, yaitu:

Kromosom [1] = [GFEDCBH]

Kromosom [2] = [DEBCHGF]

Kromosom [3] = [EDCBHGF]

Kromosom [4] = [FBCDEGH]

Kromosom [5] = [HFCDEGB]

Kromosom [6] = [CDEFGHB]

Kromosom [7] = [BHFCDEG]

Kromosom [8] = [GHBCDEF]

Kromosom [9] = [EHFDCBG]

Kromosom [10] = [GFEDCBH]

e. Mutasi

Pada kasus TSP ini, skema mutasi yang digunakan adalah *swapping mutation*. Jumlah kromosom yang mengalami mutasi dalam satu populasi ditentukan oleh

parameter *mutation rate* (*pm*). Proses mutasi dilakukan dengan cara menukar gen yang dipilih secara acak dengan gen sesudahnya. Jika gen berada di akhir kromosom, maka gen tersebut ditukar dengan gen yang pertama.

Pertama, kita hitung dulu panjang total gen yang ada pada satu populasi:

Panjang total gen = jumlah gen dalam satu kromosom * jumlah kromosom

$$= 7 * 10 = 70$$

Pemilihan posisi gen yang mengalami mutasi dilakukan dengan membangkitkan bilangan acak antara 1 sampai panjang total gen, yaitu 1 sampai 24. Misal kita tentukan $pm=20\%$. Maka jumlah gen yang akan dimutasi adalah

$$\text{Jumlah gen} = 0,2 * 70 = 14$$

Posisi tersebut didapat dari pembangkitan 14 bilangan acak 1-70. 14 buah posisi gen yang akan dimutasi setelah diacak adalah posisi 45, 57, 6, 67, 65, 43, 18, 62, 36, 52, 30, 68, 6, 39.

Jadi, populasi setelah dilakukan *mutasi*, yaitu:

Kromosom [1] = [GFEDCBH]

Kromosom [2] = [DEBCHGF]

Kromosom [3] = [EDCHBGF]

Kromosom [4] = [FBCDEGH]

Kromosom [5] = [HCFDEGB]

Kromosom [6] = [DCEGFHB]

Kromosom [7] = [HBCFDEG]

Kromosom [8] = [GHCBEDF]

Kromosom [9] = [HEFDCBG]

Kromosom [10] = [GEFCBDH]

Setelah proses mutasi selesai, maka proses Algoritma Genetika untuk 1 generasi telah selesai. Nilai *fitness* setelah 1 generasi adalah:

$$f_1 = AG + GF + FE + ED + DC + CB + BH + HA$$

$$f_1 = 7 + 5 + 14 + 10 + 7 + 18 + 13 + 10 = 84$$

$$f_2 = AD + DE + EB + BC + CH + HG + GF + FA$$

$$f_2 = 70 + 10 + 23 + 18 + 22 + 6 + 5 + 13 = 167$$

$$f_3 = AE + ED + DC + CH + HB + BG + GF + FA$$

$$f_3 = 15 + 10 + 7 + 22 + 13 + 22 + 5 + 13 = 107$$

$$f_4 = AF + FB + BC + CD + DE + EG + GH + HA$$

$$f_4 = 13 + 25 + 18 + 7 + 10 + 16 + 6 + 10 = 105$$

$$f_5 = AH + HC + CF + FD + DE + EG + GB + BA$$

$$f_5 = 10 + 22 + 13 + 56 + 7 + 16 + 22 + 35 = 184$$

$$f_6 = AB + BD + DE + EG + GF + FH + HC + CA$$

$$f_6 = 35 + 45 + 10 + 16 + 5 + 12 + 22 + 40 = 185$$

$$f_7 = AG + GH + HC + CF + FD + DE + EB + BA$$

$$f_7 = 7 + 6 + 22 + 13 + 56 + 10 + 23 + 35 = 172$$

$$f_8 = AG + GH + HC + CB + BE + ED + DF + FA$$

$$f_8 = 7 + 6 + 22 + 18 + 23 + 10 + 56 + 13 = 155$$

$$f_9 = AG + GH + HF + FD + DC + CB + BE + EA$$

$$f_9 = 7 + 6 + 12 + 56 + 7 + 18 + 23 + 15 = 144$$

$$f_{10} = AG + GE + EF + FC + CB + BD + DH + HA$$

$$f_{10} = 7 + 16 + 14 + 13 + 18 + 45 + 65 + 10 = 188$$

Sebelumnya telah ditentukan kriteria berhenti, yaitu bila setelah dalam beberapa generasi berturut-turut diperoleh nilai *fitness* terendah yang tidak berubah. Pada 1 generasi telah terlihat bahwa terdapat nilai tujuan/nilai *fitness* terkecil yang tidak berubah, yaitu $f_1 = 84$. Apabila perhitungan dilanjutkan hingga ke generasi ke-N, maka diyakinkan bahwa nilai *fitness* yang terendah tetap tidak akan berubah. Walaupun perhitungan cukup dijabarkan hingga generasi ke-1 saja, solusi yang mendekati optimal telah didapatkan.

Dari hasil perhitungan di atas, jarak terdekat yang akan ditempuh adalah 84 dengan rute perjalanan $AG \rightarrow GF \rightarrow FE \rightarrow ED \rightarrow DC \rightarrow CB \rightarrow BH \rightarrow HA$ atau sebaliknya $AH \rightarrow HB \rightarrow BC \rightarrow CD \rightarrow DE \rightarrow EF \rightarrow FG \rightarrow GA$.

F. Kesimpulan

Berdasarkan uraian di atas dapat diambil beberapa kesimpulan, yaitu:

1. Solusi yang diperoleh tergantung pada penentuan operator dan parameter yang digunakan dalam Algoritma Genetika untuk menyelesaikan masalah. Operator genetika meliputi seleksi, *crossover* dan mutasi. Parameter meliputi ukuran populasi, jumlah generasi, probabilitas *crossover* dan probabilitas mutasi.
2. Penggunaan parameter yang tepat akan mempercepat tercapainya kromosom dengan nilai *fitness* tinggi.
3. Karakteristik Algoritma Genetika adalah mencari calon-calon solusi optimal secara random sehingga Algoritma Genetika tidak selalu memberikan hasil yang terbaik, meskipun pada umumnya solusi yang diberikan mendekati hasil optimum dengan waktu yang diperlukan relatif singkat.
4. Algoritma Genetika sebagai Algoritma Heuristik dapat melakukan suatu pencarian solusi untuk masalah *Traveling Salesman Problem* (TSP) dengan nilai *fitness*-nya didefinisikan sebagai *invers* dari fungsi tujuan.

Daftar Pustaka

- Bambang, E. K., 2008, *Implementasi Algoritma Paralel Genetic Algorithm Untuk Penyelesaian Heterogeneous Fleet Vehicle Routing Problem*, Skripsi Fakultas Teknik Informasi Institut Teknologi Sepuluh Nopember, Surabaya.
- Fitria, dkk, 2005, *Penerapan Algoritma Genetika pada Persoalan Pedagang Keliling*, Makalah STIMIK ITB, Bandung.
- Herlianti, M., 2005, *Implementasi Algoritma Ant Colony Optimization untuk Traveling Salesman Problem*, Skripsi MIPA UGM, Yogyakarta.
- Munir, R., 2001, *Matematika Diskrit*, Informatika, Bandung.
- Obitko, M., 1998, *An Introduction to Genetic Algorithm*, Czech technical University, <http://cs.Felk.cvut.cz/~obitko.ga/>.
- Widhiyasa, A., 2007, *Kajian Genetic Algorithm dalam Penyelesaian TSP*, Makalah STIMIK, Bandung